NASA/CR-97- 206117

FINAL

[illegible]

[illegible]

0 8 713

FINAL REPORT

JOINT CONTRACTURE ORTHOSIS (JCO)

by

Thomas R. Lunsford, M.S.E., C.O., Principal Investigator
Ken Parsons, M.D., Investigator
Thomas Krouskop, P.E., Ph.D., Investigator
Kevin McGee, Ph.D., Investigator

#H-4852 - JOINT CONTRACTURE ORTHOSIS (JCO)

GOAL

The purpose of this project was to develop an advanced orthosis which is effective in reducing upper and lower limb contractures in significantly less time than currently required with conventional methods. The team that developed the JCO consisted of an engineer, orthotist, therapist, and physician.

INTRODUCTION

To the patient with spinal injury, joint contracture is devastating. Elbow flexion contractures prevent the patient with quadriplegia from being able to propel and transfer to and from a wheelchair. Knee flexion contractures prevent a patient with weakened lower limb muscles from walking. In short, significant functional gains are denied patients with contractures.

Normal recumbent and sitting positions promote the unwanted flexion postures of upper and lower limbs. Inexorably human joints that are not moved develop myostatic contractures. These contractures are comprised of adhesions between the intersections of collagen fibers which surround, and are intertwined with, muscle fibers. Once a myostatic contracture has developed, current reduction strategies are difficult and costly. The conventional methods of reducing contractures include such efforts as intensive physical therapy for special positioning and stretching, progressive casting techniques which involves a therapist, physician and cast-room technician services, all of which are time consuming for the personnel and a long involved process for the patient. Also, this type of management is highly dependent upon the skill and intensity of care provided by the staff.

Several orthoses have been developed to effect contracture reduction and have had limited success. The goals of this type of management are to have a method that can provide consistent positive outcome and require minimal extra patient management time.

In the past contracture reduction orthoses consisted of two cuffs held together with a mechanical joint. An elastic mechanism or a turnbuckle would produce the force to reduce the contracture. Orthoses incorporating a turnbuckle utilized the concept of constant displacement with variable force. When the elastic mechanism was used to reduce the contracture, the concept of constant force with variable displacement was being used. Others have tried casting to reduce contractures. Although successful, this approach requires many months for a desirable outcome and is labor intensive since several casts have to be applied before the joint attains compete range of motion.

METHOD

The mechanism underlying soft tissue contracture is sensitive to excessive force in that if excessive force is applied during the lengthening process, an antagonistic response can be elicited which worsens the contracture. Moreover, gross manual stretching with high forces can also be painful and produce trauma that further reduces the range of motion.

The system developed incorporates is a new type of Joint Contracture Orthosis (JCO) and involves passive lengthening of myostatic contractures at a very slow rate such that the afferent stimulus which triggers the antagonistic reflex is avoided. Silence of the antagonistic muscle group can be verified by examination or in the future can be confirmed by EMG. The amount of contracture reduction torque is clinically adjustable.

There are four parts to the envisioned (JCO): a small low voltage motor with a speed reduction gear box and screw type drive, a mechanism on the screw drive to sense torque resistance offered by the involved joint, an electronic microprocessor to operate the small motor, and a custom orthosis upon which the motor and screw drive are mounted.

The microprocessor (laptop computer) enables the motor to begin challenging the tight tissues. When the clinically determined amount of torque resistance is encountered, the motor will stop. The microprocessor will dwell at this point whereupon collagen adhesions are being separated. After approximately 2 minutes, the microprocessor reverses the motor direction for a short period of time until the tissues are slack (measured via palpation) and then the motor stops to allow homeostasis a chance to remove waste products and repair separated collagen fibers. After another dwell period with the tissues in slack, the cycle starts anew. This time, however, the joint angle at which the preset torque resistance is encountered is slightly greater (improved). The joint contracture after each cycle is reduced a fraction of a degree and after a one hour treatment the contracture is reduced several degrees. Assuming the minimum time for the four phases of the cycle is approximately four minutes, the nominal treatment time is 60 minutes or 15 cycles.

To preserve the joint contracture reduction attained, a turnbuckle is installed on the JCO; after the therapy it is used to maintain the joint angle and prevent the collagen fibers from reforming adhesive intersection bonds. The tension on the turnbuckle will be slightly less than applied in the last cycle of treatment.

Initially the knee and elbow joints on adult patients will be targeted in phase II of this project, "Clinical Trials." These joints are the most frequently encountered clinically, have relatively long lever segments, and will provide the patient with immediate functional gain. Only joint contractures with functional potential will be treated. Joint angle will be measured pre- and post-treatment with a mechanical adapted protractor attached to the sidebars of the orthosis.

DESIGN RESULTS

The design criteria for the JCO designed and fabricated were:

1. The maximum moment that the device is able to generate is 100 in-lb.
2. The moment selection increment is 1 in-lb.
3. The maximum range of motion provided by the JCO is 150 degrees.
4. The maximum speed is approximately one degree per minute in the ;powered mode.
5. The drive mechanism can be disengaged so that the orthosis can be positioned and adjusted at the start of the therapy session.
6. The therapy cycle time range is adjustable from software.
7. The power for the system is from a grounded wall socket.

The JCO is composed of four distinct components. These components are:

1. A hand-held computer for controlling the system.

2. The joint torque sensor.

3. The drive mechanism.

4. The interface circuitry.

Hand-held Computer

A hand-held, 286-compatible computer running DOS is used for controlling the JCO. The computer was purchased from Kila Systems in Boulder, Colorado. The computer is called "PC-in-a-box" and it has 128 KB of SRAM, 2 serial ports and 2 parallel ports. Due to the widespread availability of Pascal compilers and the fact that many people are familiar with the Pascal language, programming of the JCO control computer was done in Pascal. A copy of the code used to control the JCO is included in Appendix 1.

Joint Torque Sensor

The torque sensor is a product of Transducer Techniques, Inc. The part number for the product is TRT-200. In use, the output from the torque sensor was amplified and converted to a digital signal to provide the needed control input. The computer reads the analog-to-digital converter to monitor the joint torque in the JCO.

Drvie Mechanism

The drive mechanism for the JCO consists of a motor, gear system, and a mechanism to disengage the motor from the orthosis. The motor used to power the drive mechanism is a Micro-Mo motor with a planetary gearhead and integral optical shaft encoder, model 3540 K 012 C + 34 PG 23520:1 + HEM 3540 K 1. The final drive reducton is a worm gear purchased from Pic Design, MWW 6-3 (worm gear) and MWW 5-3-C (worm wheel). A mechanical slip clutch ordered from Polyclutch Division, Custom Products Corp.,h part number SAS 16, was also incorporated into the drive train so that the torque delivered to the anatomical joint by the JCO can be assured to be always less than 100 in-lb and thus minimize the possibility that the device can cause harm to the patient who is using the system for therapy.

Interface Circuitry

A variety of circuitry was designed and fabricated to interface the computer with the sensors and motor of the JCO. Schematics for the circuitry are shown in Figure 1. Switching the motor between forward and reverse directions is accompolished with a pair of relays. TTL compatible signals from the computer are amplified and then used to energize the relay coils which control the motor. The encoder on the motor is an incremental encoder which provides very precise information about the changes in angle that occur between the arms of the JCO during the therapy session; but is also is desirable to get an estimate of the absolute position of the JCO. A potlentiometer is used for this purpose. The voltage drop across the potentiomenter in response to a known current is measured, digitized, and fed to the computer as angle information.

The components for the system were assembled and the software debugged and the system was bench tested to assure that it met the design specifications. The results of the bench testing indicated that the system performed within the desired range of operation and that it should be safe to use in treating the series of experimental subjects described in the original proposal.

OPERATION

The JCO is attached to a custom fitted or custom fabricated orthosis (elbow or knee initially) . The motor and drive mechanism is attached to the sidebar of the orthosis. For application to elbow flexion or extension contractures, the orthosis will be custom fit to the patient in bed and the JCO attached and operated.

Specific Operating Procedure

To operate the JCO use the following steps:

1. Plug in the Kila PC in a Box and the control box.

2. Turn the power "on" for both pieces of equipment.

3. If the prompt does not appear on the Kila, reboot the computer by holding down the power key until the screen blacks-out.

4. Once the prompt appears on the Kila, type in "c:" and hit the enter key. The colon is entered by holding down the 2nd key while typing a semi-colon.

5. C> should appear as the prompt now.

6. Type in "master" and hit the enter key.

7. A sentence telling the current angle of the anatomical joint should be displayed by a request for the maximum torque.

8. Type in the value of the maximum allowable torque in inch-pounds and hit the enter key.

9. A request to enter the therapy routine should appear.

10. To enter the therapy routine, type in one of four commands. These commands are +, -, h, or e. The + extends the anatomical joint, the - contracts it, the h holds the joint in the current position, and the e ends the session.

11. Directly following the command enter either the number of degrees of movement for the + or -, the number of seconds to hold for the h, or 0 for e.

12. The command should appear like one of the following: +5, -3, h4, e0.

13. After the first instruction is written, hit the enter key.

14. If the instruction entered does not follow the form described above, upon execution of the instruction, the program will display the statement "Invalid instruction" and skip to the next movement in the session.

15. Repeat steps 10 through 14 until the entire therapy session has been entered into the Kila. Make sure the last instruction is e0.

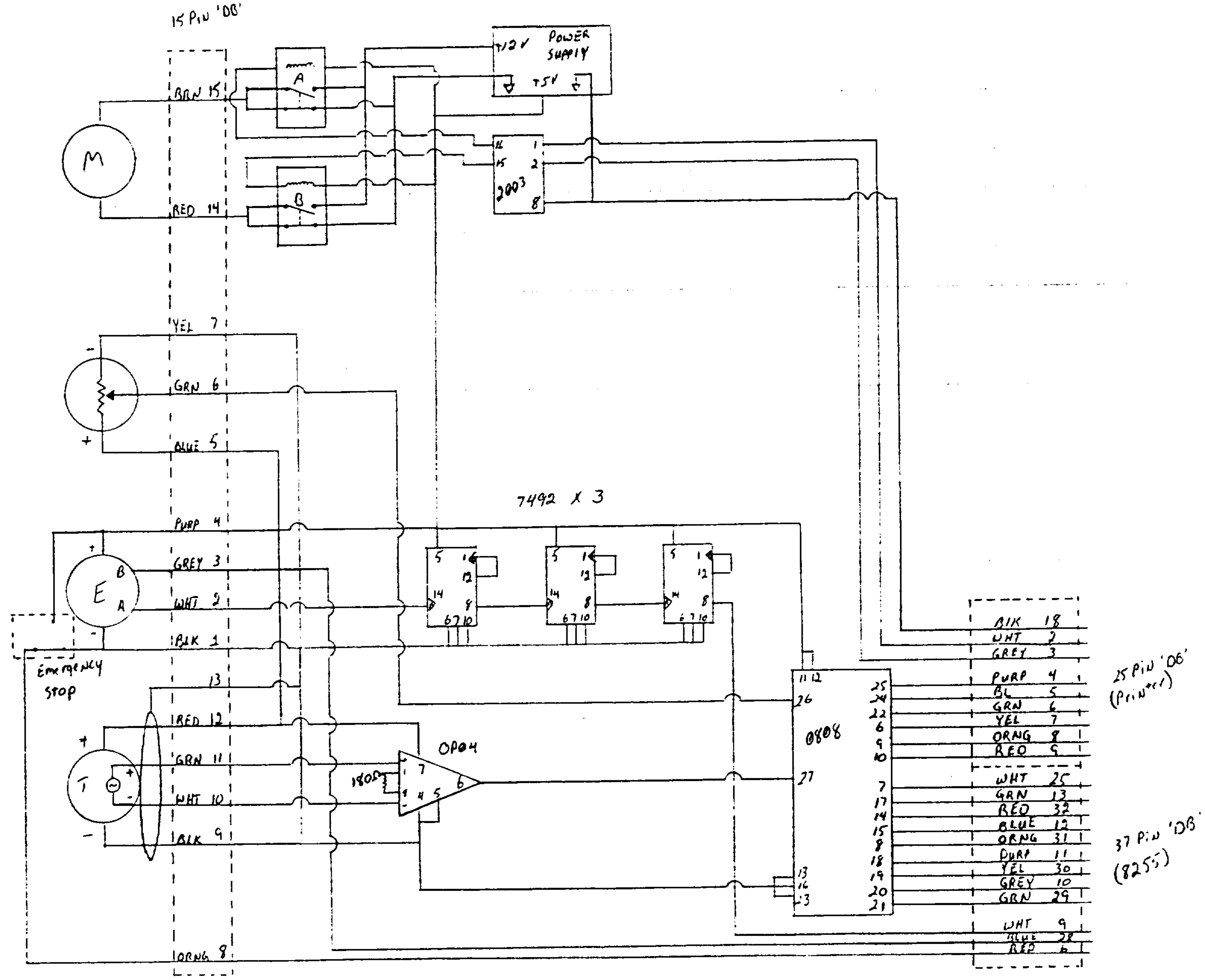
16. Once the enter key has been pressed after the e0 command, the Kila will begin to execute the therapy session.

17. The starting angle of the anatomical joint should be displayed again followed by a description of the current movement being executed.

18. If at anytime, the torque on the mechanical joint of the JCO exceeds the maximum allowable torque entered at the start of the session, the movement currently being executed will end, and the next step will begin.

19. If at anytime during the session you wish to discontinue the current motion, flip the emergency stop switch on the control box. This will create a sentence asking if you would like to continue. Enter either y or n. N will end the session, and y will move to the next step abandoning the interrupted step.

20. Once the therapy session is complete, turn off the power on the control box and the Kila, and unplug both pieces of equipment.


15 Pin 'DB'
M
BRN 15
RED 14
A
B
+12V
POWER SUPPLY
+5V
2003
11
15
1
2
8
YEL 7
GRN 6
BLUE 5
PURP 4
GREY 3
WHT 2
BLK 1
13
RED 12
GRN 11
WHT 10
BLK 9
ORNG 8
E
B
A
T
Emergency Stop
7492 X 3
5
14
1
12
8
6 7 10
OP04
180Ω
1
7
8
4
5
6
0808
11 12
26
27
13
16
13
25
24
22
6
9
10
7
17
14
15
8
18
19
20
21
BLK 18
WHT 2
GREY 3
PURP 4
BL 5
GRN 6
YEL 7
ORNG 8
RED 9
25 Pin 'DB' (Printer)
WHT 25
GRN 13
RED 32
BLUE 12
ORNG 31
PURP 11
YEL 30
GREY 10
GRN 29
WHT 9
BLUE 28
RED 6
37 Pin 'DB' (8255)

APPENDIX 1

```
(*$include:'ibmport.int'*)

program master (input,output);

uses ibmport;

label 10, 1000;

const
 {888 = printer port; 616 = sensor port; 617 = port b}
 {619 = control port; 3 = stop; 2 = forward; 1 = reverse}
 x0 = 1;
 x1 = 2;
 x2 = 4;
 x3 = 8;
 x4 = 16;
 x5 = 32;
 x6 = 64;
 x7 = 128;

type
  table = array [0..255] of word;
  motion = record
           sign : char;
           degree : integer;
          end;

var
 torque,maximum,move: word;
 j,items,s,m,old,now,tmp: integer;
 check,choice: char;
 g,pot,d0,d1,max: word;
 info: array [1..255] of motion;
 count: integer4;
 temp: real8;
 p: table;

function calculate (k,q,l,n,m,p,z:word):word;
 {computes the value to be sent to the printer port}

 begin  {calculate}
  calculate :=
    (k*x0)+(q*x1)+(l*x2)+(n*x4)+(m*x5)+(p*x6)+(z*x7);
 end;   {calculate}

function maxtorque (x: word): word;
 {computes maximum torque value for A2D from inch-ounces}

begin  {maxtorque}
 maxtorque := x * 2;
end;   {maxtorque}

procedure degrees (var x: table);
```

```
 {sets up potentiometer conversion table}

var i: integer;

begin   {degrees}
 for i := 0 to 40 do x[i] := 58;
 x[41] := 59; x[42] := 60; x[43] := 60; x[44] := 61;
 x[45] := 61; x[46] := 62; x[47] := 62; x[48] := 63;
 x[49] := 63; x[50] := 64; x[51] := 64; x[52] := 65;
 x[53] := 66; x[54] := 66; x[55] := 67; x[56] := 68;
 x[57] := 68; x[58] := 69; x[59] := 70; x[60] := 70;
 x[61] := 71; x[62] := 72; x[63] := 73; x[64] := 73;
 x[65] := 74; x[66] := 74; x[67] := 75; x[68] := 75;
 x[69] := 76; x[70] := 76; x[71] := 77; x[72] := 78;
 x[73] := 78; x[74] := 79; x[75] := 80; x[76] := 80;
 x[77] := 81; x[78] := 81; x[79] := 82; x[80] := 82;
 x[81] := 83; x[82] := 84; x[83] := 85; x[84] := 85;
 x[85] := 86; x[86] := 86; x[87] := 87; x[88] := 88;
 x[89] := 88; x[90] := 89; x[91] := 89; x[92] := 90;
 x[93] := 90; x[94] := 91; x[95] := 91; x[96] := 92;
 x[97] := 92; x[98] := 93; x[99] := 94; x[100] := 94;
 x[101] := 95; x[102] := 96; x[103] := 96; x[104] := 97;
 x[105] := 98; x[106] := 98; x[107] := 99; x[108] := 100;
 x[109] := 100; x[110] := 101; x[111] := 101; x[112] := 102;
 x[113] := 102; x[114] := 103; x[115] := 103; x[116] := 104;
 x[117] := 104; x[118] := 105; x[119] := 105; x[120] := 106;
 x[121] := 107; x[122] := 108; x[123] := 108; x[124] := 109;
 x[125] := 109; x[126] := 110; x[127] := 110; x[128] := 111;
 x[129] := 112; x[130] := 112; x[131] := 113; x[132] := 114;
 x[133] := 114; x[134] := 115; x[135] := 115; x[136] := 116;
 x[137] := 117; x[138] := 118; x[139] := 119; x[140] := 120;
 x[141] := 120; x[142] := 121; x[143] := 121; x[144] := 122;
 x[145] := 122; x[146] := 123; x[147] := 124; x[148] := 124;
 x[149] := 125; x[150] := 125; x[151] := 126; x[152] := 126;
 x[153] := 127; x[154] := 128; x[155] := 129; x[156] := 130;
 x[157] := 130; x[158] := 131; x[159] := 132; x[160] := 132;
 x[161] := 133; x[162] := 133; x[163] := 134; x[164] := 134;
 x[165] := 135; x[166] := 135; x[167] := 136; x[168] := 138;
 x[169] := 138; x[170] := 139; x[171] := 139; x[172] := 140;
 x[173] := 140; x[174] := 141; x[175] := 141; x[176] := 142;
 x[177] := 143; x[178] := 144; x[179] := 144; x[180] := 145;
 x[181] := 146; x[182] := 146; x[183] := 147; x[184] := 148;
 x[185] := 148; x[186] := 149; x[187] := 150; x[188] := 151;
 x[189] := 152; x[190] := 152; x[191] := 153; x[192] := 153;
 x[193] := 154; x[194] := 154; x[195] := 155; x[196] := 156;
 x[197] := 156; x[198] := 157; x[199] := 158; x[200] := 158;
 x[201] := 159; x[202] := 160; x[203] := 160; x[204] := 161;
 x[205] := 162; x[206] := 162; x[207] := 163; x[208] := 164;
 x[209] := 164; x[210] := 165; x[211] := 165; x[212] := 166;
 x[213] := 166; x[214] := 167; x[215] := 168; x[216] := 168;
 x[217] := 169; x[218] := 170; x[219] := 170; x[220] := 171;
 x[221] := 172; x[222] := 172; x[223] := 173; x[224] := 174;
 x[225] := 175; x[226] := 175; x[227] := 176; x[228] := 176;
 x[229] := 177; x[230] := 178; x[231] := 178; x[232] := 179;
 x[233] := 180; x[234] := 180; x[235] := 180; x[236] := 181;
 x[237] := 182; x[238] := 182; x[239] := 182; x[240] := 183;
 x[241] := 183; x[242] := 183;
 for i := 243 to 255 do x[i] := 184;
end;    {degrees}

procedure clock (var x,k,q,l,n,m,p,z:word);
 {clocks the ADC0808 chip}

 begin  {clock}
  z := 1;
  x := calculate(k,q,l,n,m,p,z);
  outp (888,x);
  z := 0;
  x := calculate(k,q,l,n,m,p,z);
  outp (888,x);
 end;   {clock}

procedure hold (x: integer; var y : integer);
 {waits for desired amount of time}

 var
  i,k,h,: integer;
  counter : integer4;
  g : word;
  choice : char;

 begin   {hold}
  counter := 0;
  if (x > 0) then begin {if}
   for i := 1 to x do
    for k := 1 to 103 do
     for h := 1 to 200 do begin {for}
      counter := counter + 1;
      g := inp(617);
      m := ord (g);
      for s := 0 to 5 do
       m := m div 2;
      m := m mod 2;
      if (m = 1) then begin {if}
       writeln ('Do you wish to continue?');
       readln (choice);
       if (choice = 'Y') or (choice = 'y') then begin {if}
        y := y + 1;
        goto 10;
        end
       else goto 1000;
       end;  {if}
     end;  {for}
  end;    {if}
 end;     {hold}
```

```
function sensor (x: char; d0,d1: word):word;
 {reads torque or potentiometer}

var
 a,b,i,k,d,d2,d4,d5,d6,d7: word;

 begin   {sensor}
  if (x = 'p') then d2 := 0 else d2 := 1;
  for k := 1 to 3 do begin
   d4 := 0;
   d5 := 0;
   d6 := 0;
   d7 := 0;
   d := calculate(d0,d1,d2,d4,d5,d6,d7);
   outp (888,d);
   clock (d,d0,d1,d2,d4,d5,d6,d7);
   clock (d,d0,d1,d2,d4,d5,d6,d7);
   d4 := 1;          {address latch enable}
   d := calculate(d0,d1,d2,d4,d5,d6,d7);
   outp (888,d);
   clock (d,d0,d1,d2,d4,d5,d6,d7);
   d5 := 1;          {start}
   d := calculate(d0,d1,d2,d4,d5,d6,d7);
   outp (888,d);
   clock (d,d0,d1,d2,d4,d5,d6,d7);
   d4 := 0;
   d5 := 0;
   d := calculate(d0,d1,d2,d4,d5,d6,d7);
   outp (888,d);
   for i := 1 to 65 do
    clock (d,d0,d1,d2,d4,d5,d6,d7);
   d6 := 1;
   d := calculate(d0,d1,d2,d4,d5,d6,d7);
   outp (888,d);
   clock (d,d0,d1,d2,d4,d5,d6,d7);
   clock (d,d0,d1,d2,d4,d5,d6,d7);
   a := inp (616);
  end;    {for}
 sensor := a;
 end;   {sensor}

begin   {main}
 outp (619,146);
 degrees (p);
 check := 'p';
 pot := sensor (check,d0,d1);
 tmp := ord (pot);
 move := p[tmp];
 writeln ('Now at ',move:3,' degrees');
 d0 := 1;
 d1 := 1;
 outp (888,3);
 count := 0;
 temp := 0;
 old := 0;
 now := 0;
 writeln ('Enter maximum torque');
 readln (maximum);
 max := maxtorque (maximum);
 writeln ('Enter therapy routine');
 items:=0;
j:=0;
repeat
j:=j+1;
items:=items+1;
read(info[j].sign);
readln(info[j].degree);
until(info[j].sign = 'E') or (info[j].sign = 'e');
j:=1;
torque := 0;
check := 'p';
pot := sensor (check,d0,d1);
tmp := ord (pot);
move := p[tmp];
writeln ('Starting at ',move:3,' degrees');
10 : repeat
  if (info[j].sign = '+') then begin {positive}
   temp := info[j].degree * 13.6;   {calculate count}
   count := round4(temp);
   d0 := 0;
   d1 := 1;
   outp (888,2);
   writeln ('Extending ',info[j].degree:3,' degrees');
   while (count > 0) and (torque < max) do begin {while}
    g := inp (617);
    m := ord (g);
    now := m mod 2;
    if (now = 0) and (now <> old) then count := count - 1;
    old := now;
    check := 't';
    torque := sensor (check,d0,d1);
    g := inp(617);
    m := ord (g);
    for s := 0 to 5 do
     m := m div 2;
    m := m mod 2;
    if (m = 1) then begin  {if}
     d0 := 1;
     d1 := 1;
     outp (888,3);
     writeln ('Do you want to continue?');
     readln (choice);
     if (choice = 'y') or (choice = 'Y') then begin  {if}
      j := j + 1;
      goto 10;
      end {if}
   else goto 1000;
   end;  {if}
 end;   {while}
 d0 := 1;
 d1 := 1;
 outp (888,3);
 j := j + 1;
 end  {positive}
else if (info[j].sign = '-') then begin  {negative}
 temp := (info[j].degree) * 13.6;  {compute count}
 count := abs(round4(temp));
 d0 := 1;
 d1 := 0;
 outp (888,1);
 writeln ('Contracting ',info[j].degree:3,' degrees');
 while (count > 0) do begin  {while}
  g := inp (617);
  m := ord (g);
  now := m mod 2;
  if (now = 0) and (now <> old) then count := count - 1;
  old := now;
  g := inp(617);
  m := ord (g);
  for s := 0 to 5 do
   m := m div 2;
  m := m mod 2;
  if (m = 1) then begin  {if}
   d0 := 1;
   d1 := 1;
   outp (888,3);
   writeln ('Do you wish to continue?');
   readln (choice);
   if (choice = 'Y') or (choice = 'y') then begin {if}
    j := j + 1;
    goto 10;
    end
   else goto 1000;
   end;   {if}
 end;    {while}
 d0 := 1;
 d1 := 1;
 outp (888,3);
 j := j + 1;
 end  {negative}
else if (info[j].sign = 'H') or (info[j].sign = 'h') then
 begin  {hold}
  d0 := 1;
  d1 := 1;
  outp (888,3);
  writeln ('Holding ',info[j].degree:3,' seconds');
  hold (info[j].degree,j);
  j := j + 1;
 end   {hold}
```

```
  else if ((info[j].sign <> 'E') and (info[j].sign <> 'e'))
   then begin {if}
    writeln ('Invalid instruction');
    j := j + 1;
    end;  {if}
 until j = items;
1000 : writeln ('Therapy is done!');
end.     {main}
```